23/11/2021

GMT351 GEOSPATIAL DATA MANAGEMENT

Midterm Project Report

ABDULSAMET TOPTAŞ – 21905024

INTRODUCTION

This report includes the diagram of the virtual marketing I designed for the Spatial data management midterm project, and the stages of designing it with queries on PostgreSQL.

DATABASE DESIGN

First of all, I would like to introduce the work I will design in PostgreSQL on the diagram. I wanted to take the virtual markets that we use almost every day in our daily life as a subject. As we can see on the diagram in "Figure 1" that I shared below when starting my project, we have 6 main headings and cardinalities that relate these main headings. If I start to briefly introduce the relations of the project, our markets have employees, so I connected the market and employees with "has". I have specified this relationship as " 1-N", that is, a market can have more than one employee, but employees cannot work in more than one market. Unlike that, I associated products and campaigns as "1-1". That is, a product can have a campaign, and a campaign cannot contain more than one product. I completed my diagram with similar logic in other relationships and moved on to the design phase in PostgreSQL.

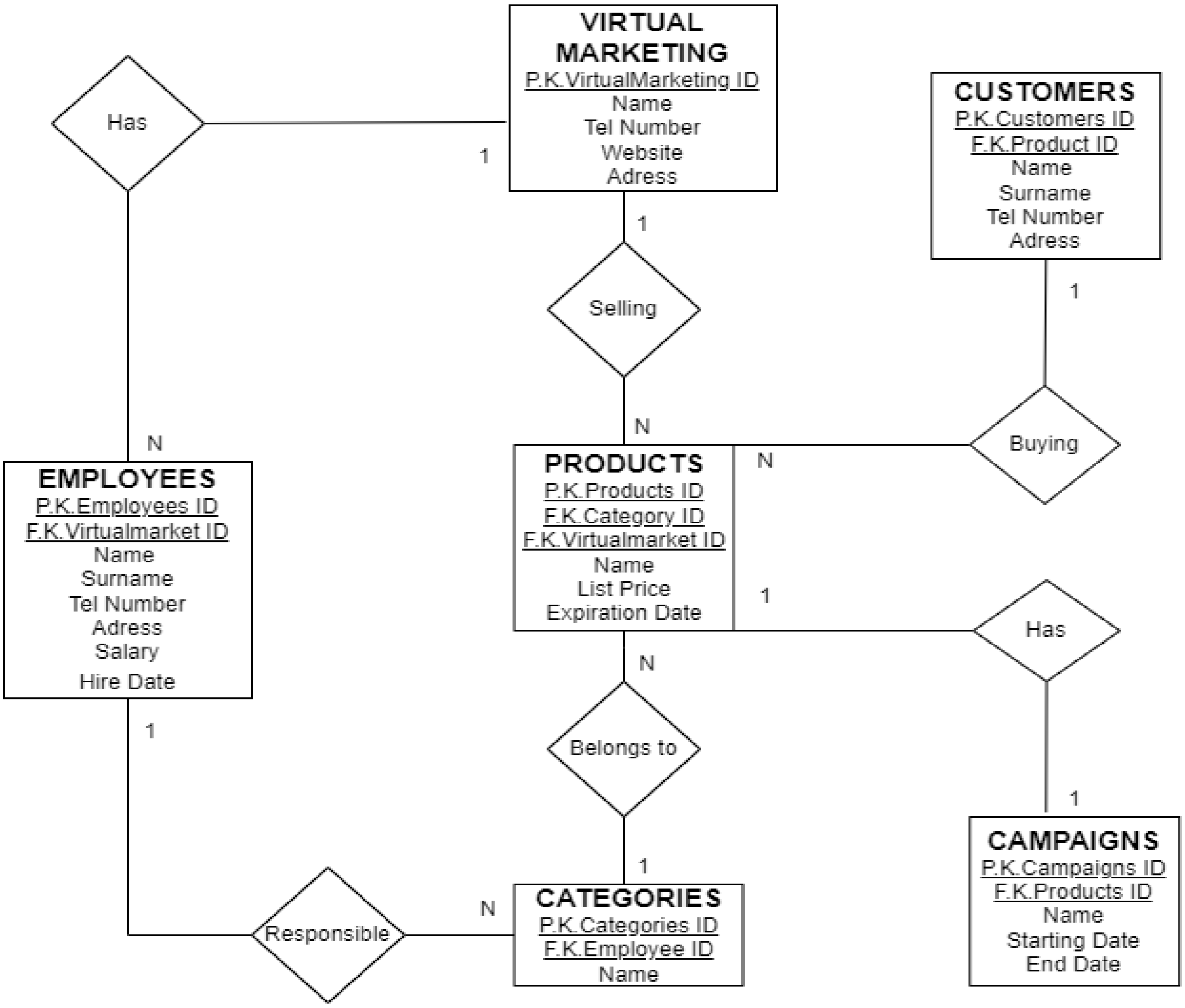
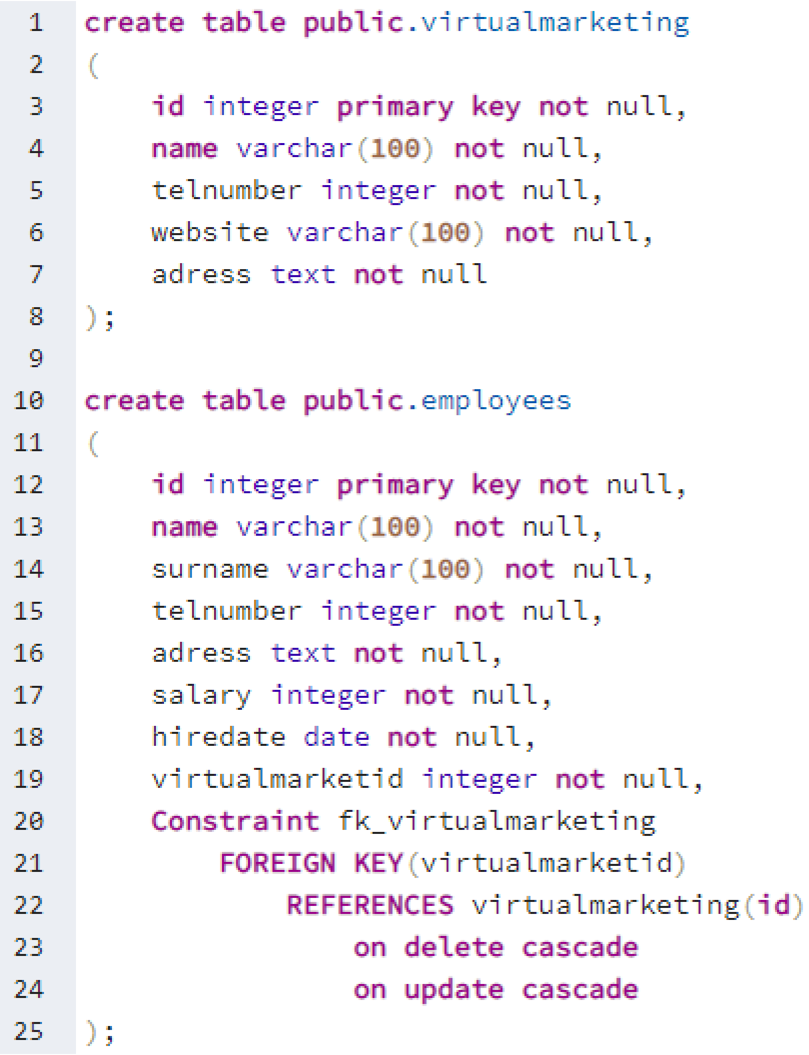


Figure 1

POSTGRESQL DESIGN

1- Creating Table

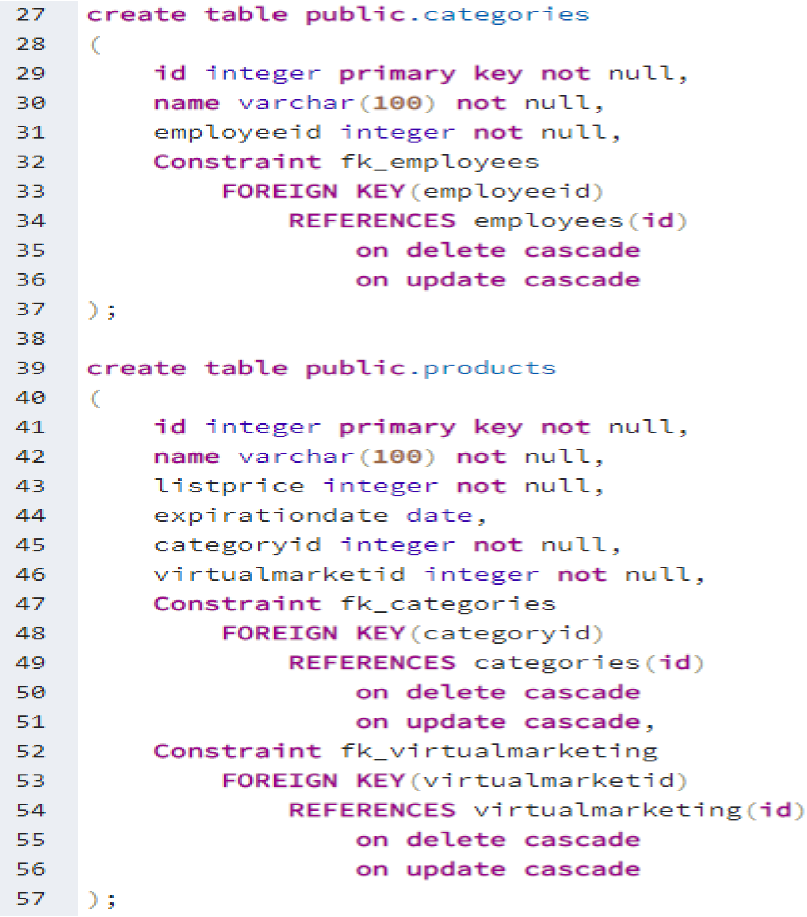
At this stage, I opened the section where I would write my codes from the "Query Tool" tab in PostgreSQL and started creating my tables.



```
1   create table public.virtualmarketing
2   (
3       id integer primary key not null,
4       name varchar(100) not null,
5       telnumber integer not null,
6       website varchar(100) not null,
7       adress text not null
8   );
9
10  create table public.employees
11  (
12      id integer primary key not null,
13      name varchar(100) not null,
14      surname varchar(100) not null,
15      telnumber integer not null,
16      adress text not null,
17      salary integer not null,
18      hiredate date not null,
19      virtualmarketid integer not null,
20      Constraint fk_virtualmarketing
21          FOREIGN KEY(virtualmarketid)
22              REFERENCES virtualmarketing(id)
23                  on delete cascade
24                  on update cascade
25  );
```

Figure 2

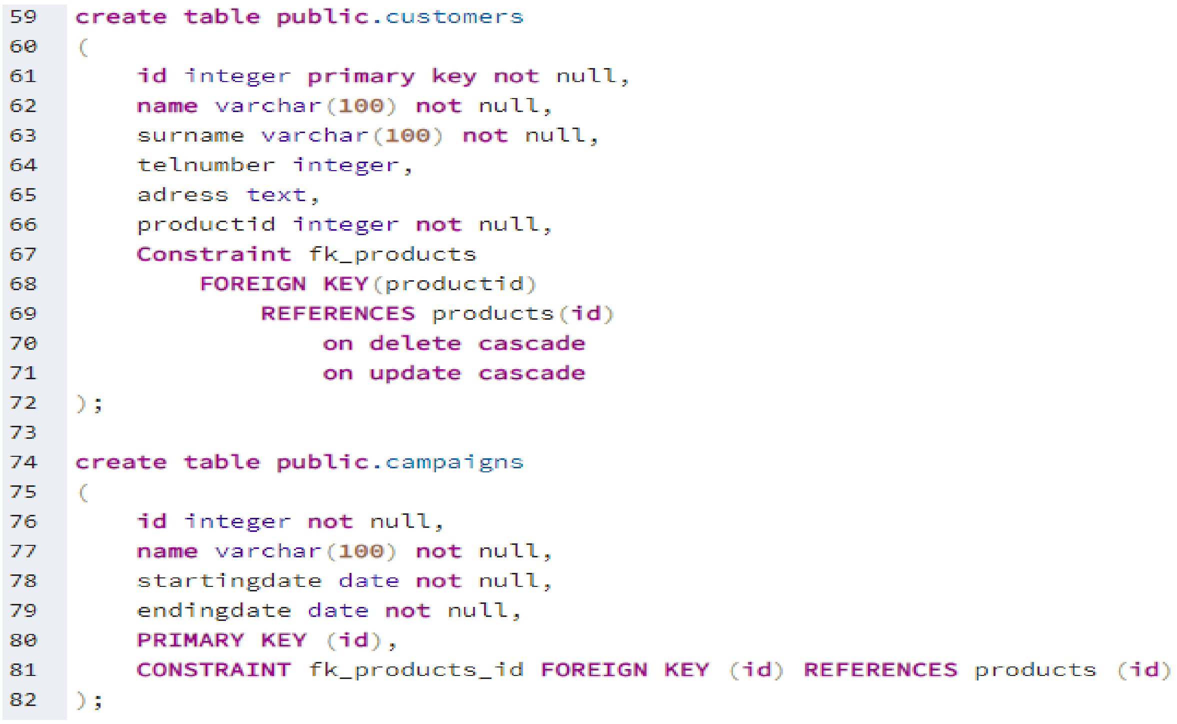
I created a Primary Key in the "Virtualmarketing" table and assigned reference information such as name, telnumber, website, address of the markets. The "id" that we assigned as the "Primary Key" allows us to uniquely identify each row in the table. With the "varchar()" command, I made the tabs such as Name and website not accept unnecessary information and set a character limit. I used the " text " command in the address and enabled us to enter our words without any limit. I used "Not Null" in all my entries, so I specified that these tabs I had opened cannot be left blank. If we examine the employees table in "Figure 2", the value I assigned as "id" is the Primary Key of the "employees" table. In this table, unlike the "Virtualmarketing" table, I added a "virtualmarketid" tab, and I assigned this tab as the "Foreign Key", thanks to the codes we saw in the 20th and 25th rows in "Figure 2", and thus, I associated the "Virtualmarketing" and " Employees" tables. Also, I assigned the codes we see in lines 20 and 25 in "Figure 2" for "1-N" relationships.



```
create table public.categories
(
    id integer primary key not null,
    name varchar(100) not null,
    employeeid integer not null,
    Constraint fk_employees
        FOREIGN KEY(employeeid)
            REFERENCES employees(id)
                on delete cascade
                on update cascade
);

create table public.products
(
    id integer primary key not null,
    name varchar(100) not null,
    listprice integer not null,
    expirationdate date,
    categoryid integer not null,
    virtualmarketid integer not null,
    Constraint fk_categories
        FOREIGN KEY(categoryid)
            REFERENCES categories(id)
                on delete cascade
                on update cascade,
    Constraint fk_virtualmarketing
        FOREIGN KEY(virtualmarketid)
            REFERENCES virtualmarketing(id)
                on delete cascade
                on update cascade
);
```

Figure 3

I applied the steps I explained above for the tables in "Figure 3" and started to create my last two tables.



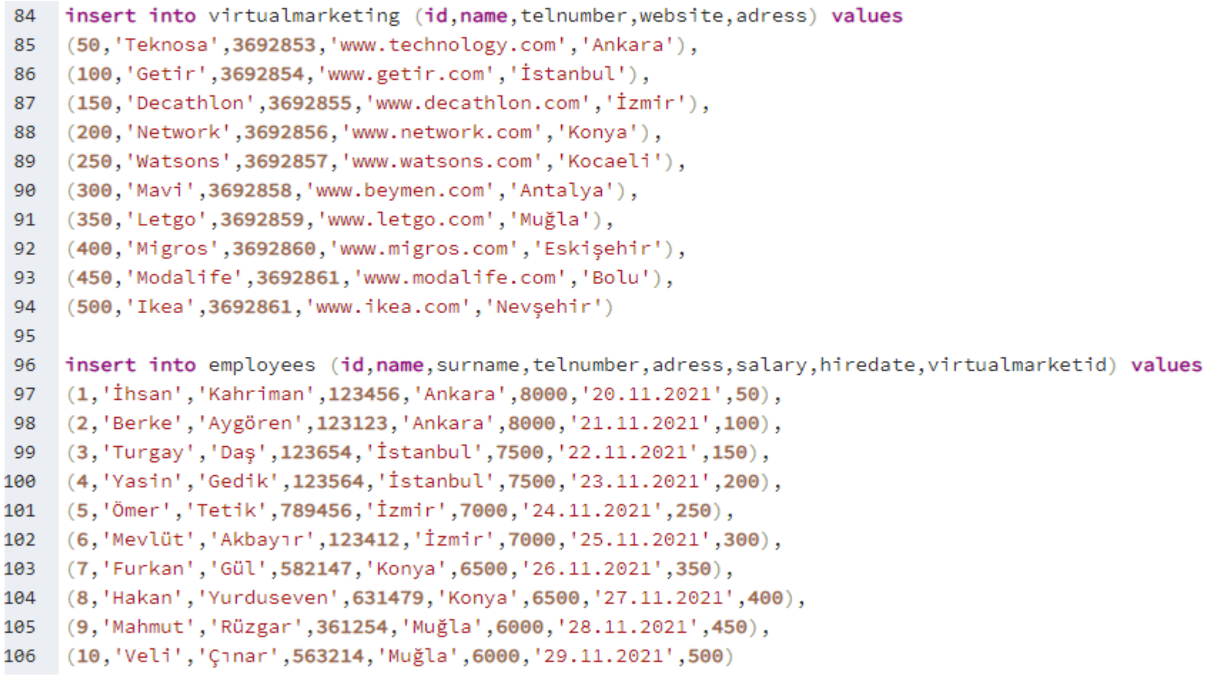
```
59  create table public.customers
60  (
61      id integer primary key not null,
62      name varchar(100) not null,
63      surname varchar(100) not null,
64      telnumber integer,
65      adress text,
66      productid integer not null,
67      Constraint fk_products
68          FOREIGN KEY(productid)
69              REFERENCES products(id)
70                  on delete cascade
71                  on update cascade
72  );
73
74  create table public.campaigns
75  (
76      id integer not null,
77      name varchar(100) not null,
78      startingdate date not null,
79      endingdate date not null,
80      PRIMARY KEY (id),
81      CONSTRAINT fk_products_id FOREIGN KEY (id) REFERENCES products (id)
82  );
```

Figure 4

There is a difference in the campaigns table in "Figure 4". Here, I assigned the codes we see in the 80th and 82nd line intervals for "1-1" relationships.

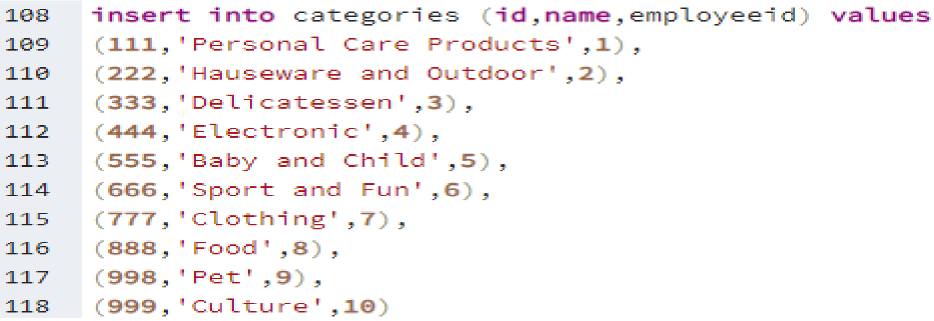
2- Adding Data to Tables

By using the "insert into" command, I added the "At least 10" entries specified for the project to the tabs I opened in the tables, using random information.



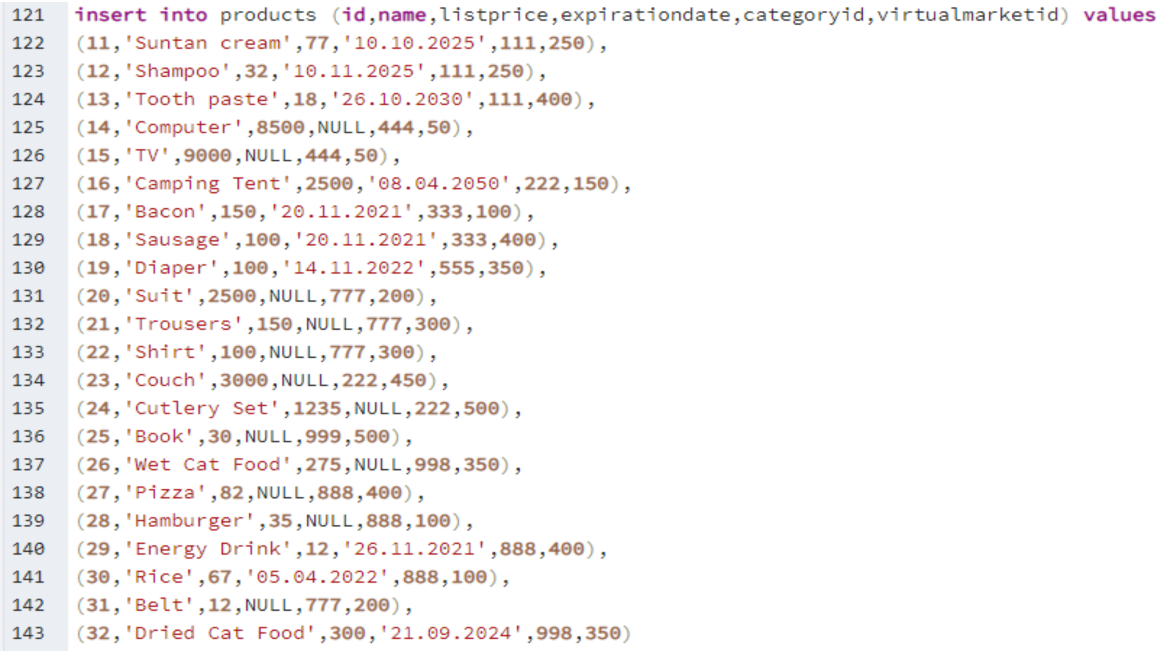
```
insert into virtualmarketing (id,name,telnumber,website,adress) values
(50,'Teknosa',3692853,'www.technology.com','Ankara'),
(100,'Getir',3692854,'www.getir.com','İstanbul'),
(150,'Decathlon',3692855,'www.decathlon.com','İzmir'),
(200,'Network',3692856,'www.network.com','Konya'),
(250,'Watsons',3692857,'www.watsons.com','Kocaeli'),
(300,'Mavi',3692858,'www.beymen.com','Antalya'),
(350,'Letgo',3692859,'www.letgo.com','Muğla'),
(400,'Migros',3692860,'www.migros.com','Eskişehir'),
(450,'Modalife',3692861,'www.modalife.com','Bolu'),
(500,'Ikea',3692861,'www.ikea.com','Nevşehir')

insert into employees (id,name,surname,telnumber,adress,salary,hiredate,virtualmarketid) values
(1,'İhsan','Kahriman',123456,'Ankara',8000,'20.11.2021',50),
(2,'Berke','Aygören',123123,'Ankara',8000,'21.11.2021',100),
(3,'Turgay','Daş',123654,'İstanbul',7500,'22.11.2021',150),
(4,'Yasin','Gedik',123564,'İstanbul',7500,'23.11.2021',200),
(5,'Ömer','Tetik',789456,'İzmir',7000,'24.11.2021',250),
(6,'Mevlüt','Akbayır',123412,'İzmir',7000,'25.11.2021',300),
(7,'Furkan','Gül',582147,'Konya',6500,'26.11.2021',350),
(8,'Hakan','Yurduseven',631479,'Konya',6500,'27.11.2021',400),
(9,'Mahmut','Rüzgar',361254,'Muğla',6000,'28.11.2021',450),
(10,'Veli','Çınar',563214,'Muğla',6000,'29.11.2021',500)
```

Figure 5



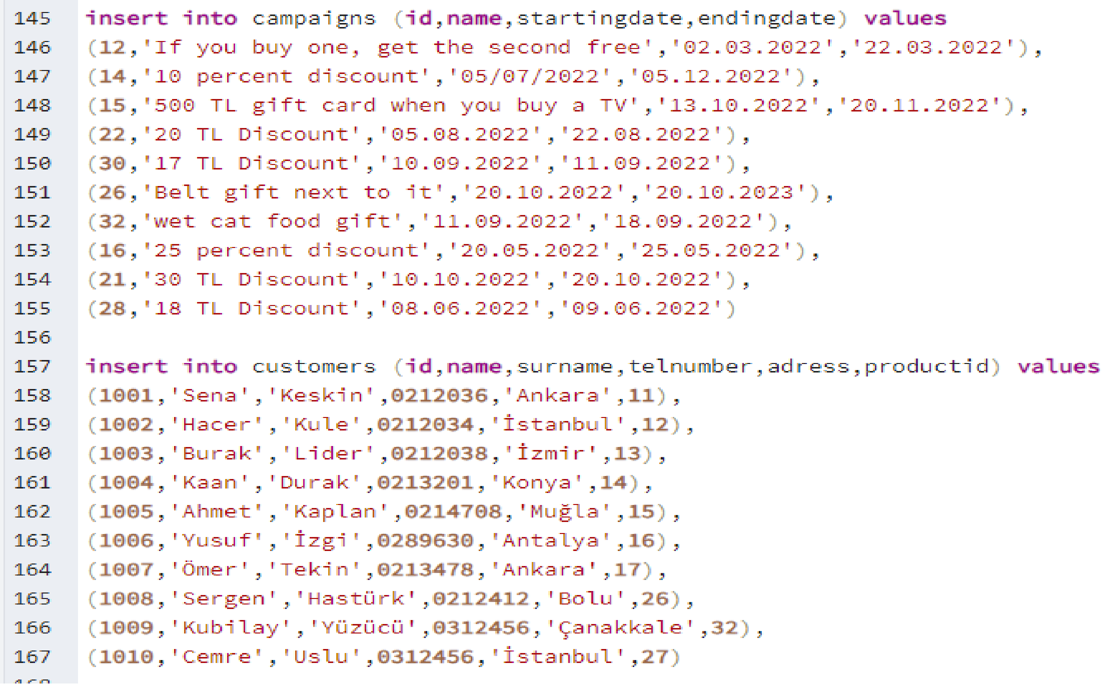
```sql
insert into categories (id,name,employeeid) values
(111,'Personal Care Products',1),
(222,'Hauseware and Outdoor',2),
(333,'Delicatessen',3),
(444,'Electronic',4),
(555,'Baby and Child',5),
(666,'Sport and Fun',6),
(777,'Clothing',7),
(888,'Food',8),
(998,'Pet',9),
(999,'Culture',10)
```

Figure 6



```sql
insert into products (id,name,listprice,expirationdate,categoryid,virtualmarketid) values
(11,'Suntan cream',77,'10.10.2025',111,250),
(12,'Shampoo',32,'10.11.2025',111,250),
(13,'Tooth paste',18,'26.10.2030',111,400),
(14,'Computer',8500,NULL,444,50),
(15,'TV',9000,NULL,444,50),
(16,'Camping Tent',2500,'08.04.2050',222,150),
(17,'Bacon',150,'20.11.2021',333,100),
(18,'Sausage',100,'20.11.2021',333,400),
(19,'Diaper',100,'14.11.2022',555,350),
(20,'Suit',2500,NULL,777,200),
(21,'Trousers',150,NULL,777,300),
(22,'Shirt',100,NULL,777,300),
(23,'Couch',3000,NULL,222,450),
(24,'Cutlery Set',1235,NULL,222,500),
(25,'Book',30,NULL,999,500),
(26,'Wet Cat Food',275,NULL,998,350),
(27,'Pizza',82,NULL,888,400),
(28,'Hamburger',35,NULL,888,100),
(29,'Energy Drink',12,'26.11.2021',888,400),
(30,'Rice',67,'05.04.2022',888,100),
(31,'Belt',12,NULL,777,200),
(32,'Dried Cat Food',300,'21.09.2024',998,350)
```

Figure 7



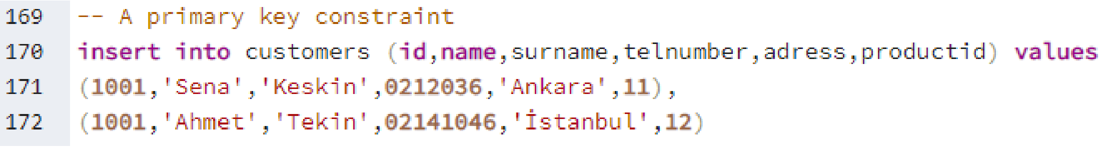
```sql
insert into campaigns (id,name,startingdate,endingdate) values
(12,'If you buy one, get the second free','02.03.2022','22.03.2022'),
(14,'10 percent discount','05/07/2022','05.12.2022'),
(15,'500 TL gift card when you buy a TV','13.10.2022','20.11.2022'),
(22,'20 TL Discount','05.08.2022','22.08.2022'),
(30,'17 TL Discount','10.09.2022','11.09.2022'),
(26,'Belt gift next to it','20.10.2022','20.10.2023'),
(32,'wet cat food gift','11.09.2022','18.09.2022'),
(16,'25 percent discount','20.05.2022','25.05.2022'),
(21,'30 TL Discount','10.10.2022','20.10.2022'),
(28,'18 TL Discount','08.06.2022','09.06.2022')

insert into customers (id,name,surname,telnumber,adress,productid) values
(1001,'Sena','Keskin',0212036,'Ankara',11),
(1002,'Hacer','Kule',0212034,'İstanbul',12),
(1003,'Burak','Lider',0212038,'İzmir',13),
(1004,'Kaan','Durak',0213201,'Konya',14),
(1005,'Ahmet','Kaplan',0214708,'Muğla',15),
(1006,'Yusuf','İzgi',0289630,'Antalya',16),
(1007,'Ömer','Tekin',0213478,'Ankara',17),
(1008,'Sergen','Hastürk',0212412,'Bolu',26),
(1009,'Kubilay','Yüzücü',0312456,'Çanakkale',32),
(1010,'Cemre','Uslu',0312456,'İstanbul',27)
```

Figure 8

3- Violated Examples

A Primary Key Constraint

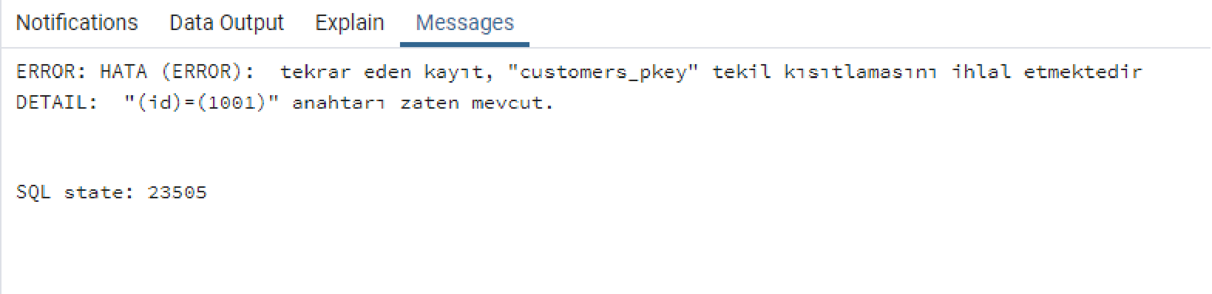
The code I wrote for the "Primary Key" violation is shown in "Figure 9". As an example here, I entered 2 different people with the same value in the "ID", which is the "Primary Key" in the "Customers" Table.



```
-- A primary key constraint
insert into customers (id,name,surname,telnumber,adress,productid) values
(1001,'Sena','Keskin',0212036,'Ankara',11),
(1001,'Ahmet','Tekin',02141046,'İstanbul',12)
```

Figure 9

I got the error "Figure 10" violates singular constraint. This is because imaginary characters have the same "ID" value.

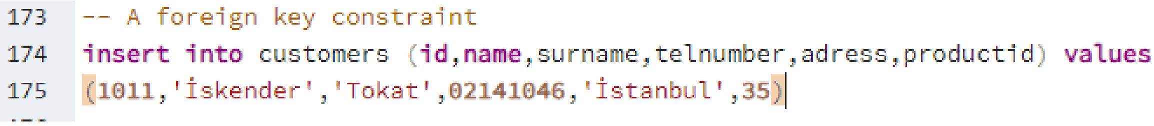


```
ERROR: HATA (ERROR):  tekrar eden kayıt, "customers_pkey" tekil kısıtlamasını ihlal etmektedir
DETAIL:  "(id)=(1001)" anahtarı zaten mevcut.


SQL state: 23505
```

Figure 10

A Foreign Key Constraint

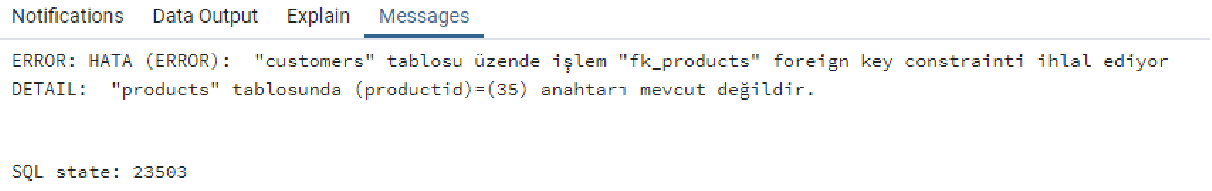
The code I wrote for the "Foreign Key" violation is shown in "Figure 11". Here, for example, I entered a product that is not in the "Products" table in the "productid" section, which I set as "Foreign Key" in the "Customers" Table.



```
-- A foreign key constraint
insert into customers (id,name,surname,telnumber,adress,productid) values
(1011,'İskender','Tokat',02141046,'İstanbul',35)
```

Figure 11

As we can see in "Figure 12", I got the error that the foreign key violates the constraint because I entered a product that is not in the table.



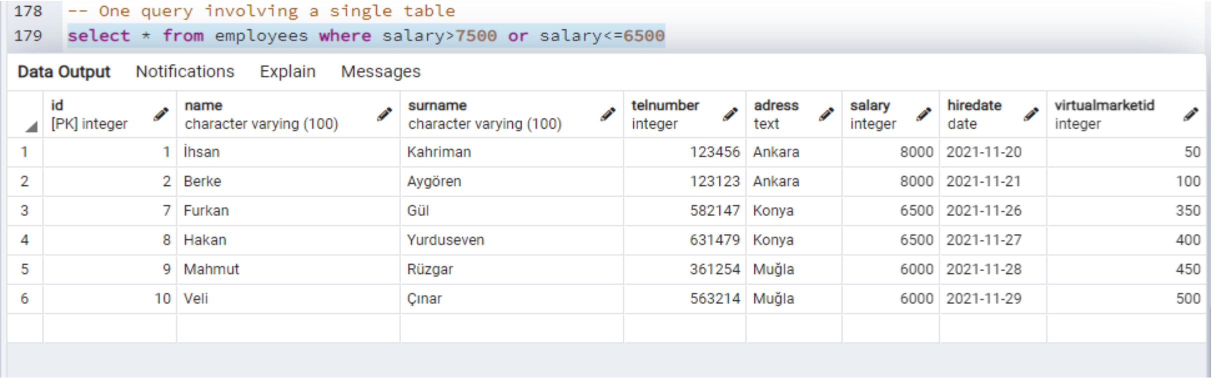
```
ERROR: HATA (ERROR):  "customers" tablosu üzende işlem "fk_products" foreign key constrainti ihlal ediyor
DETAIL:  "products" tablosunda (productid)=(35) anahtarı mevcut değildir.


SQL state: 23503
```

Figure 12

4- SQL Queries and Explanations

One Query Involving A Sıgnle Table

I have done 1 query and 1 table query using the "Select from where" commands.If I need to explain this code that I wrote in "Figure 13", it allows it to return all the information for the table I will select with " select *".I indicated which table the fields I would bring with the "From" command belonged to, and filtered the list that I would bring with the "where" command.In other words, with the command "salary>7500" and "salary<=6500", I have listed employees whose salary is greater than 7500 or whose salary is less than 6500 or equal to 6500 from all entries in the "Employees" table.



```
-- One query involving a single table
select * from employees where salary>7500 or salary<=6500
```

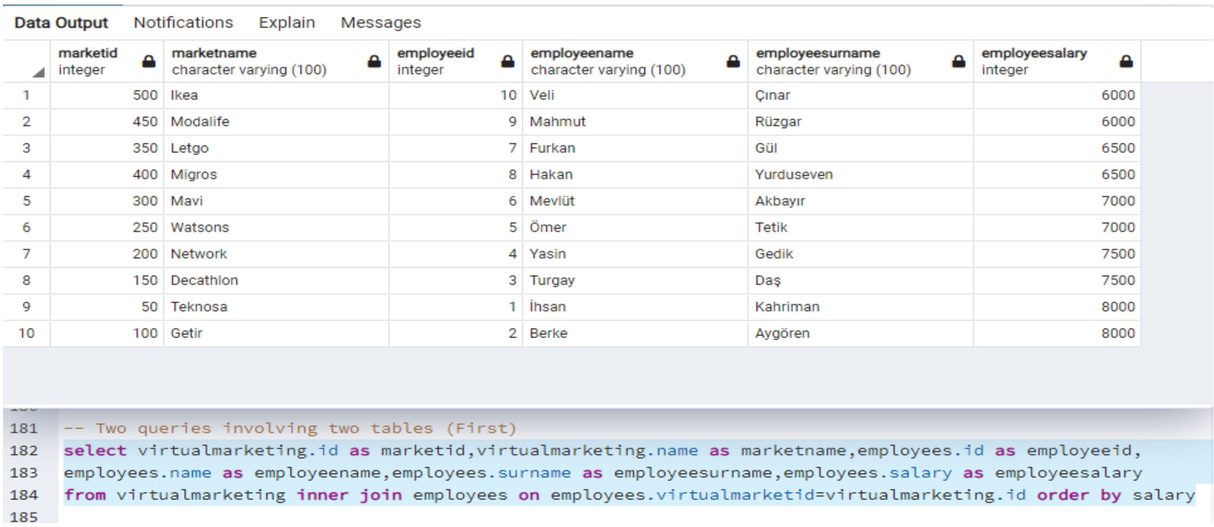
Data Output Notifications Explain Messages

	id [PK] integer	name character varying (100)	surname character varying (100)	telnumber integer	adress text	salary integer	hiredate date	virtualmarketid integer
1	1	İhsan	Kahriman	123456	Ankara	8000	2021-11-20	50
2	2	Berke	Aygören	123123	Ankara	8000	2021-11-21	100
3	7	Furkan	Gül	582147	Konya	6500	2021-11-26	350
4	8	Hakan	Yurduseven	631479	Konya	6500	2021-11-27	400
5	9	Mahmut	Rüzgar	361254	Muğla	6000	2021-11-28	450
6	10	Veli	Çınar	563214	Muğla	6000	2021-11-29	500

Figure 13

Two Queries Involving Two Tables

For two queries and two tables, I used an "inner join" command. The "Inner Join" command was used to join the relations of these two tables we selected. As seen in "Figure 14" and "Figure 15", I wrote what I would list after "select" and specified what name they would appear in the list with the "as ...." command. After the "inner join" command, I synchronize the tables that I will join with "on" with the help of "Foreign Key" and "Primary Key" and get the intersection of the tables. If we examine the output in "Figure 14", we see which employee earns how much in which market.



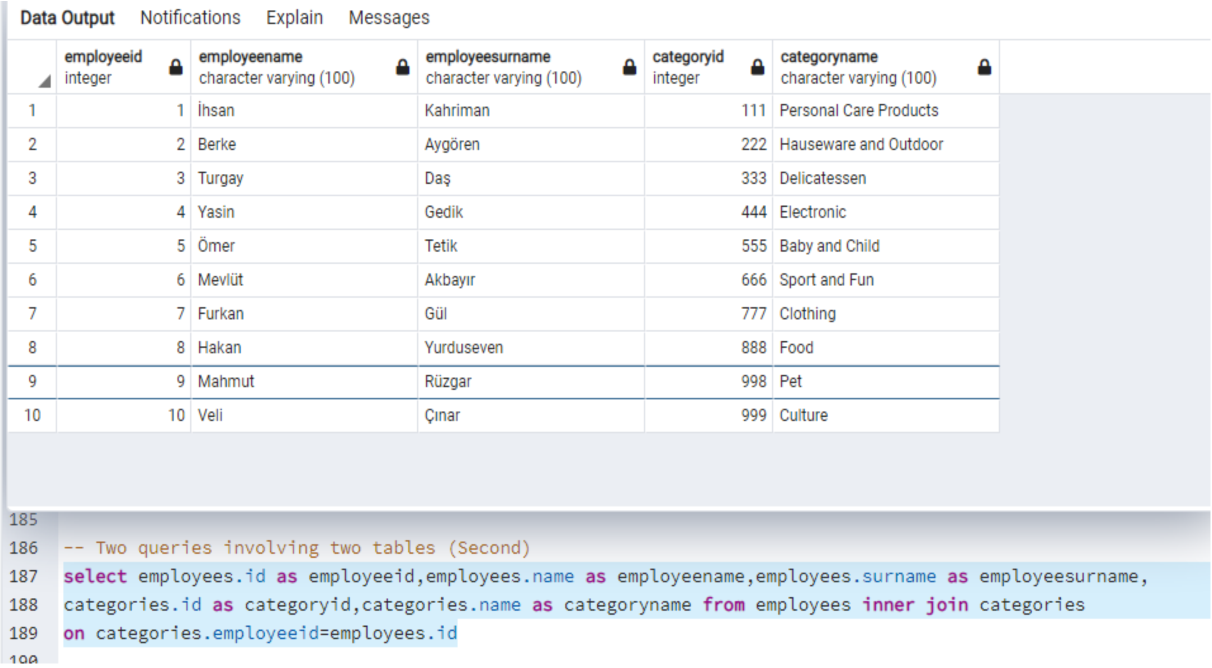
Data Output Notifications Explain Messages

	marketid integer	marketname character varying (100)	employeeid integer	employeename character varying (100)	employeesurname character varying (100)	employeesalary integer
1	500	Ikea	10	Veli	Çınar	6000
2	450	Modalife	9	Mahmut	Rüzgar	6000
3	350	Letgo	7	Furkan	Gül	6500
4	400	Migros	8	Hakan	Yurduseven	6500
5	300	Mavi	6	Mevlüt	Akbayır	7000
6	250	Watsons	5	Ömer	Tetik	7000
7	200	Network	4	Yasin	Gedik	7500
8	150	Decathlon	3	Turgay	Daş	7500
9	50	Teknosa	1	İhsan	Kahriman	8000
10	100	Getir	2	Berke	Aygören	8000

```
-- Two queries involving two tables (First)
select virtualmarketing.id as marketid,virtualmarketing.name as marketname,employees.id as employeeid,
employees.name as employeename,employees.surname as employeesurname,employees.salary as employeesalary
from virtualmarketing inner join employees on employees.virtualmarketid=virtualmarketing.id order by salary
```

Figure 14

I wrote the code in "Figure 15" with the logic in "Figure 14". If we examine the output in "Figure 15", I have found the result of which employee works in which category. Here I have combined the "Employees" and "Category" tables.



	employeeid integer	employeename character varying (100)	employeesurname character varying (100)	categoryid integer	categoryname character varying (100)
1	1	İhsan	Kahriman	111	Personal Care Products
2	2	Berke	Aygören	222	Hauseware and Outdoor
3	3	Turgay	Daş	333	Delicatessen
4	4	Yasin	Gedik	444	Electronic
5	5	Ömer	Tetik	555	Baby and Child
6	6	Mevlüt	Akbayır	666	Sport and Fun
7	7	Furkan	Gül	777	Clothing
8	8	Hakan	Yurduseven	888	Food
9	9	Mahmut	Rüzgar	998	Pet
10	10	Veli	Çınar	999	Culture

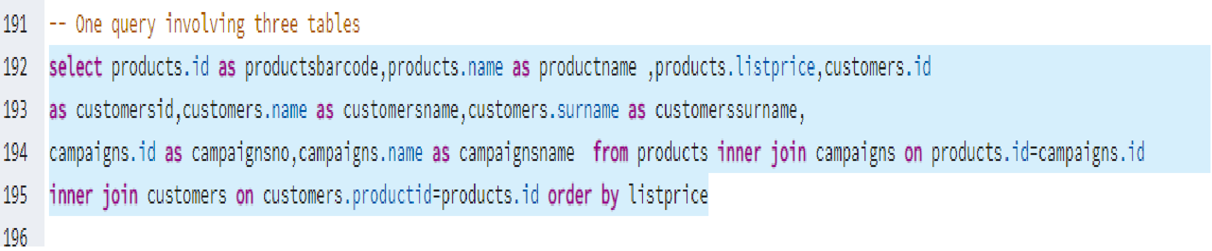
```sql

-- Two queries involving two tables (Second)
select employees.id as employeeid,employees.name as employeename,employees.surname as employeesurname,
categories.id as categoryid,categories.name as categoryname from employees inner join categories
on categories.employeeid=employees.id

```

Figure 15

One Query Involving Three Tables

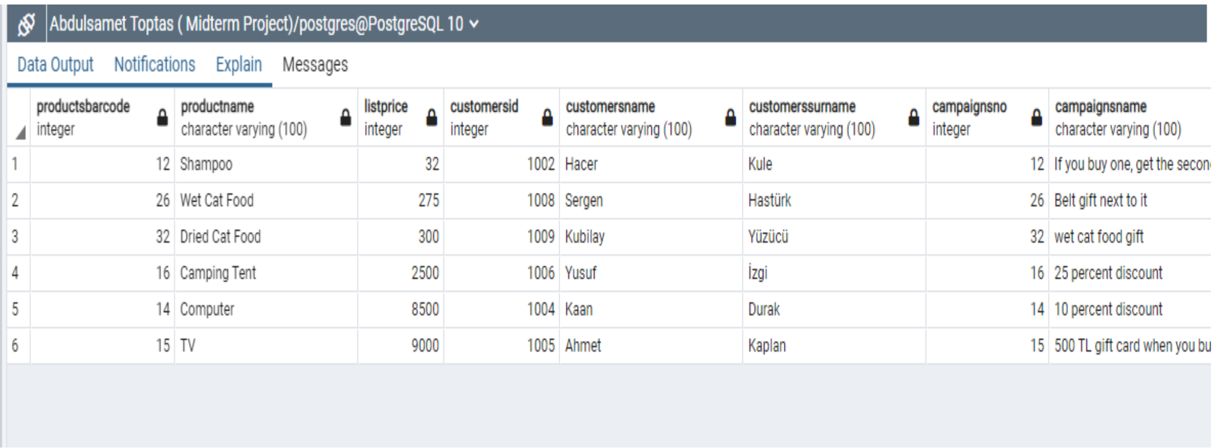
At this stage, I used the " Inner Join" command again, I think the best way to join multiple tables is the " Inner Join" command. Here, I used this command to get the intersection of "Product", "Customers" and "Campaigns" tables and was successful. First, I equated "productid" with "campaignsid" and intersected the two tables. Later, I synced the Foreign Key of "product" and "campaigns", which I linked as "1-1", to "productid" that I entered in the "Customers" table, and combined the three tables.

Let's examine "Figure 17", as the first of the three tables I combined, we see the information we want in the "Products" table, then the information of the "Customers" table, and finally the information of the "campaigns" table.



```sql
-- One query involving three tables
select products.id as productsbarcode,products.name as productname ,products.listprice,customers.id
as customersid,customers.name as customersname,customers.surname as customerssurname,
campaigns.id as campaignsno,campaigns.name as campaignsname  from products inner join campaigns on products.id=campaigns.id
inner join customers on customers.productid=products.id order by listprice

```

Figure 16



	productsbarcode integer	productname character varying (100)	listprice integer	customersid integer	customersname character varying (100)	customerssurname character varying (100)	campaignsno integer	campaignsname character varying (100)
1	12	Shampoo	32	1002	Hacer	Kule	12	If you buy one, get the secon
2	26	Wet Cat Food	275	1008	Sergen	Hastürk	26	Belt gift next to it
3	32	Dried Cat Food	300	1009	Kubilay	Yüzücü	32	wet cat food gift
4	16	Camping Tent	2500	1006	Yusuf	İzgi	16	25 percent discount
5	14	Computer	8500	1004	Kaan	Durak	14	10 percent discount
6	15	TV	9000	1005	Ahmet	Kaplan	15	500 TL gift card when you bu

Figure 17